A Simple Theory of Complex Skill Acquisition

Frank J. Lee
fjl@rpi.edu

Dept. Cognitive Science
Rensselaer Polytechnic Inst.

Niels A. Taatgen
niels@ai.rug.nl

Artificial Intelligence
University of Groningen

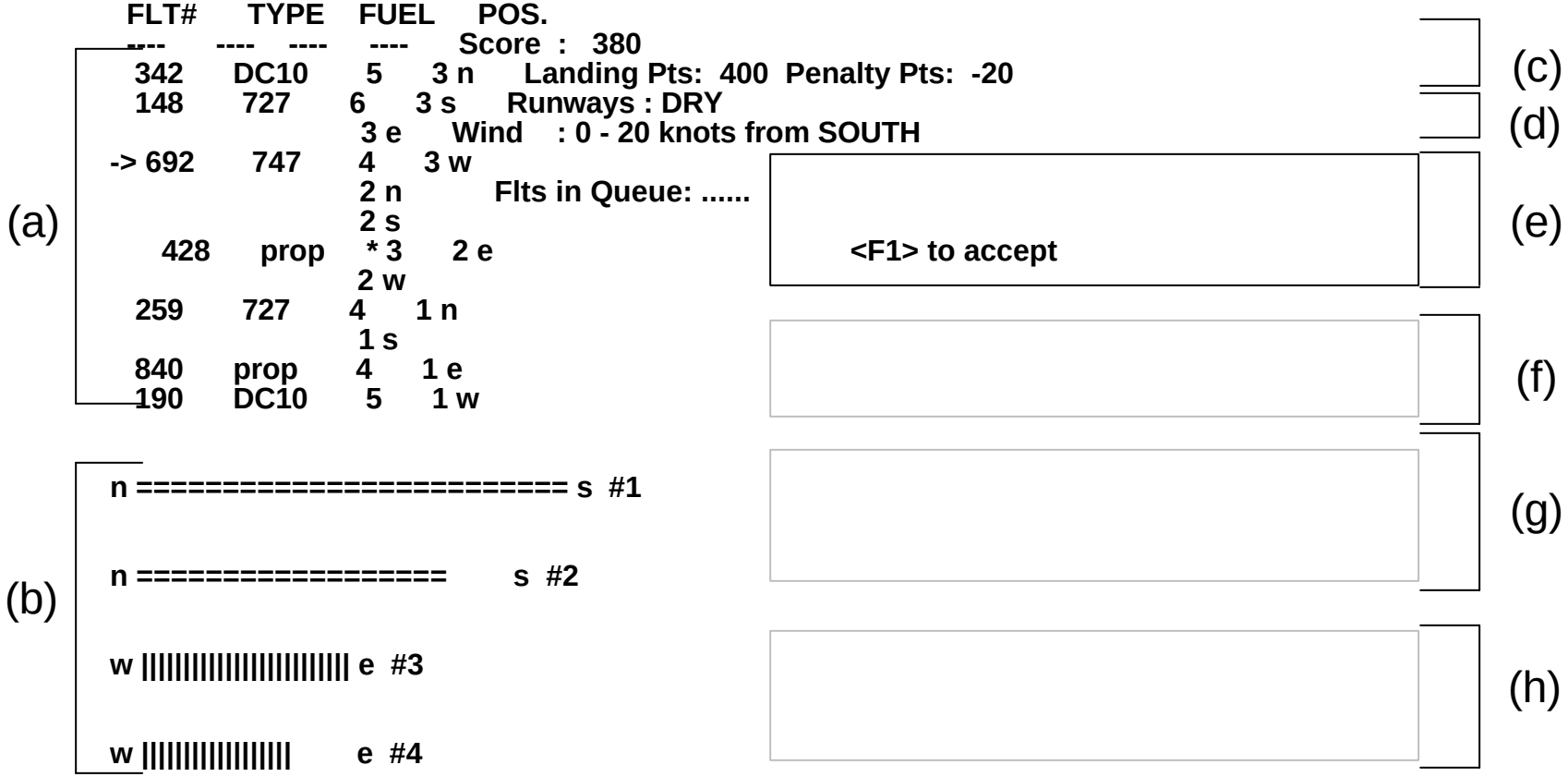
Outline

- Act I: The Data
  - A brief review of Lee & Anderson (2001)
- Act II: The Model
  - Review of past models
  - The current model
- Act III: The Conclusion
  - The model's strengths
  - The model's shortcomings

The Data

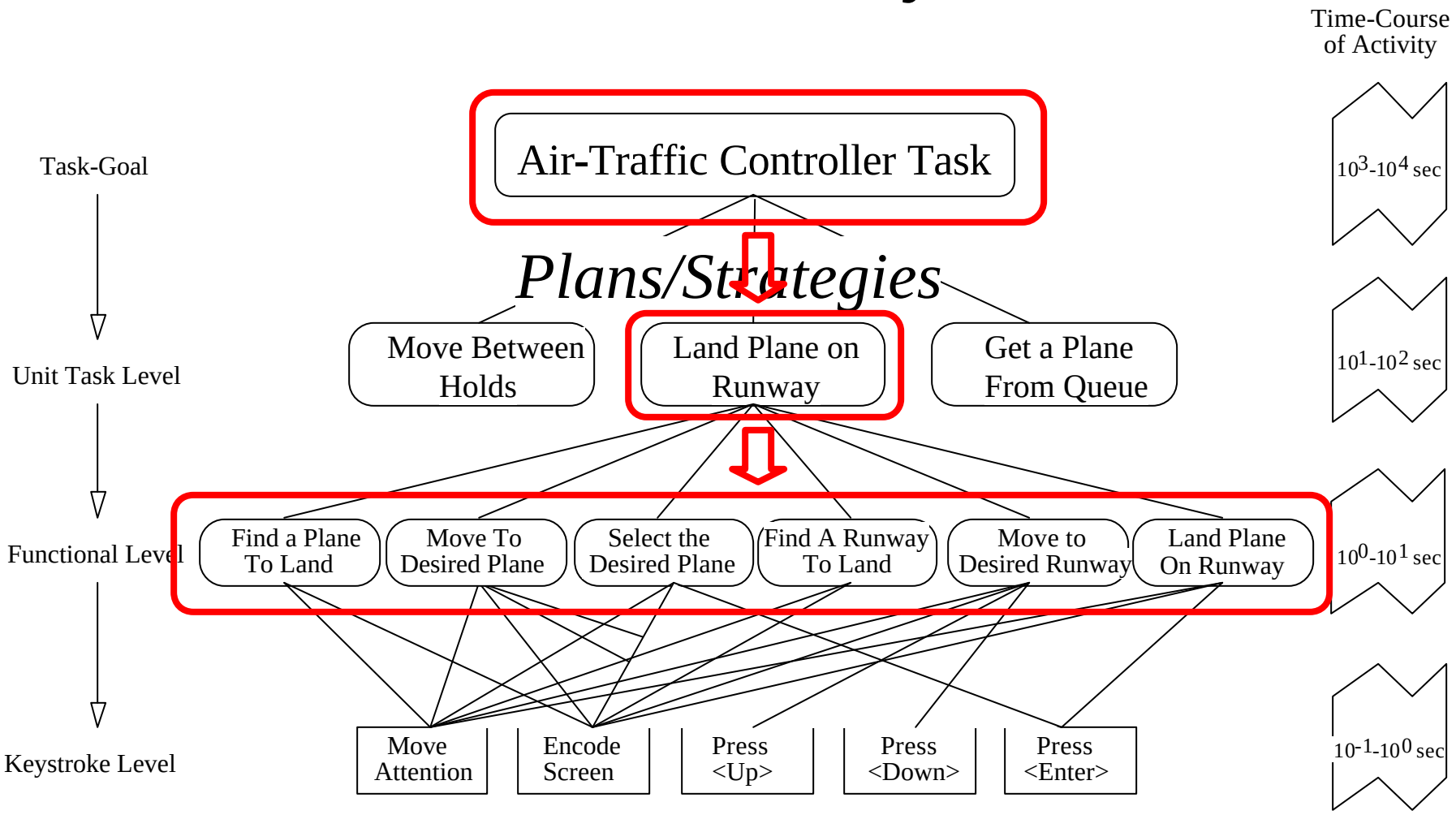
Act I

The Task

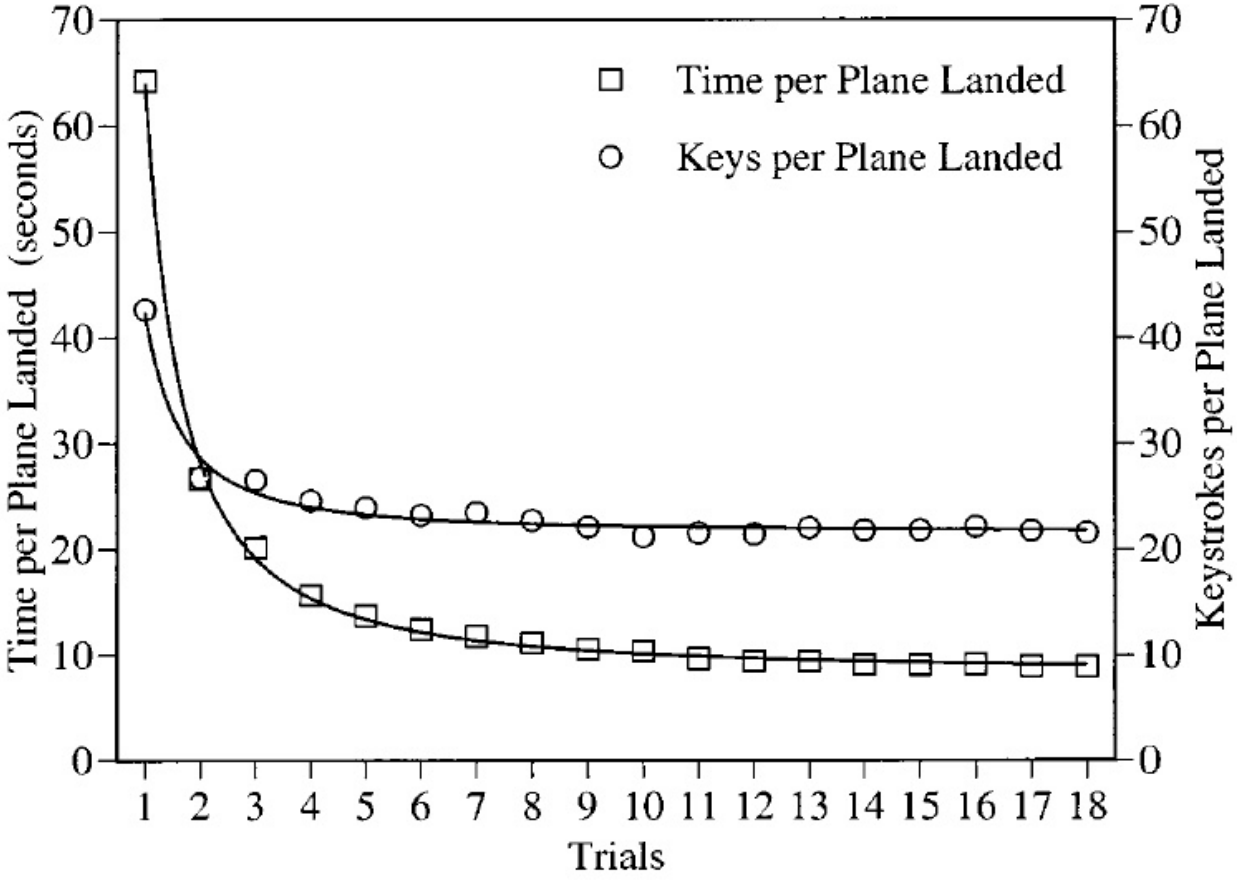


(a) Hold Positions (in 3 levels)
(b) Runways
(c) Performance Feedback
(d) Weather Information
(e) Queue
(f) Weather Change Message
(g) Error Message
(h) Rule Message

The Task Analysis

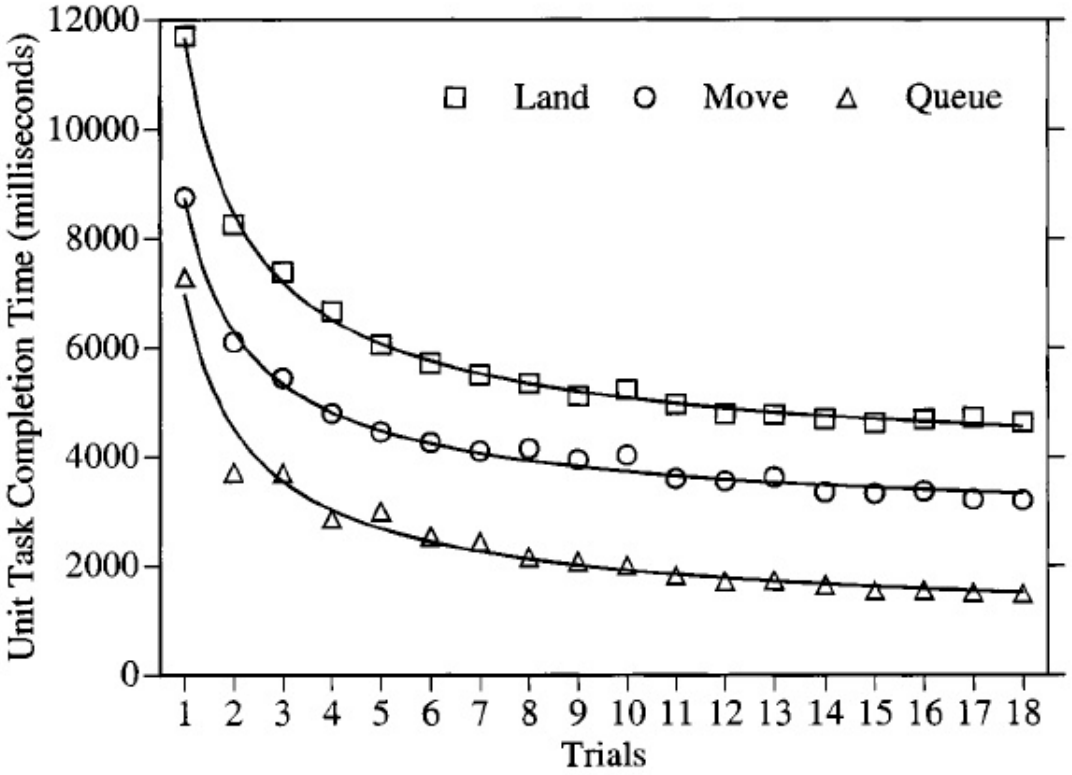


Task Level


Time per Plane Landed
Keys per Plane Landed
Time per Plane Landed (seconds)
Keystrokes per Plane Landed
Trials
0
10
20
30
40
50
60
70
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

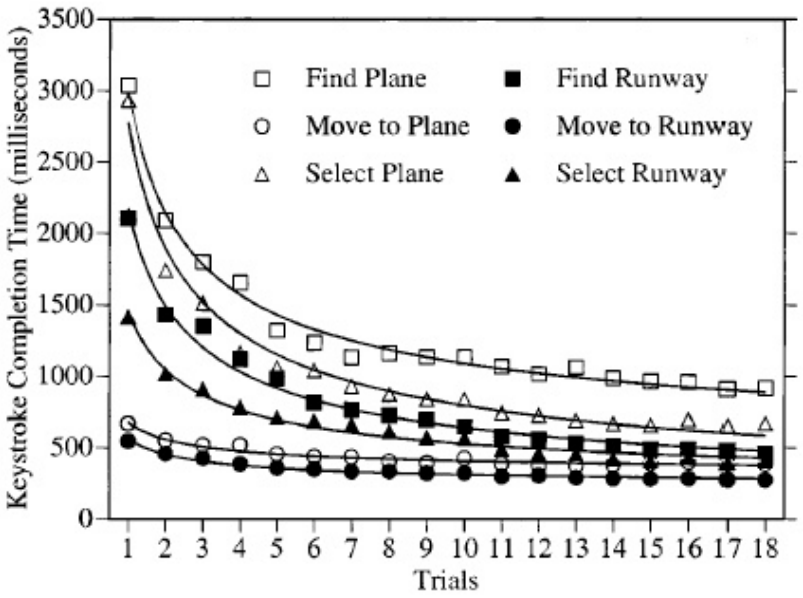
Unit Task Level

Unit-Tasks	Power Functions	R^2	χ^2	S.E.
Land Unit-Task	$T = 3582 + 8091\ N^{-0.733}$	0.997	5.061	187
Move Unit-Task	$T = 2590 + 6144\ N^{-0.733}$	0.991	13.040	147
Queue Unit-Task	$T = 768 + 6215\ N^{-0.733}$	0.971	26.544	189

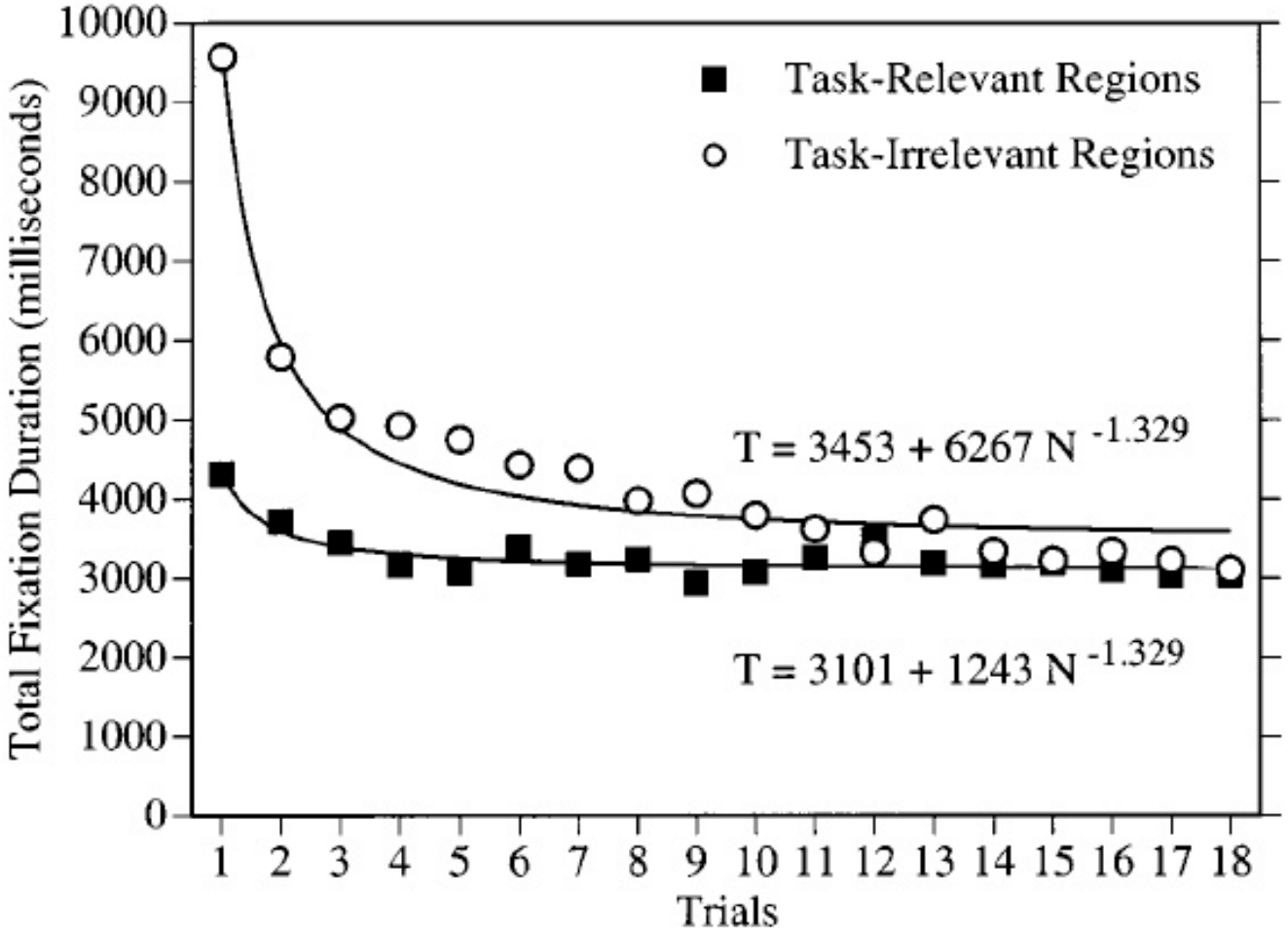


Functional Level

Unit-Tasks	Power Functions	R^2	χ^2	S.E.
Find Plane	$T = 373 + 2609\ N^{-0.561}$	0.988	9.178	79
Move to Plane	$T = 304 + 375\ N^{-0.561}$	0.954	12.786	20
Select Plane	$T = 50 + 2720\ N^{-0.561}$	0.983	20.105	69
Find Runway	$T = 61 + 2111\ N^{-0.561}$	0.985	15.453	56
Move to Runway	$T = 216 + 349\ N^{-0.561}$	0.978	16.919	11
Select Runway	$T = 175 + 128\ N^{-0.561}$	0.982	47.047	22
**Select Runway*	$T = 50 + 1378\ N^{-0.457}$	*0.982*	*29.546*	22



Where are people looking (attending)?



Conclusions Drawn From Data Analysis

- *Reducibility of Complex Tasks*: Complex tasks consist of simple keystroke level components that are being learned according to basic learning principles

- *Attentional Learning Hypothesis*: Large portion of the learning at the keystroke level reflects the learning even at a lower level, i.e. attention level, of people learning where not to look

The Model

Act II

Production Learning: A Brief History

- The Ugly Duckling: The production learning mechanism in the ACT-R theory has always been the most volatile.
  - Remember our dalliance with dependencies?
- Yet It Is Critical: To model learning in complex tasks, a good mechanism for production rule learning is vital!

- Past Is The Future: In an ironic twist, Taatgen proposed *Production Composition* (or *Proceduralization*, depending on when you ask him), a fresh take on Anderson's (1982) procedural learning theory (with a touch of Soar learning theory thrown in).

The Model: Early Years

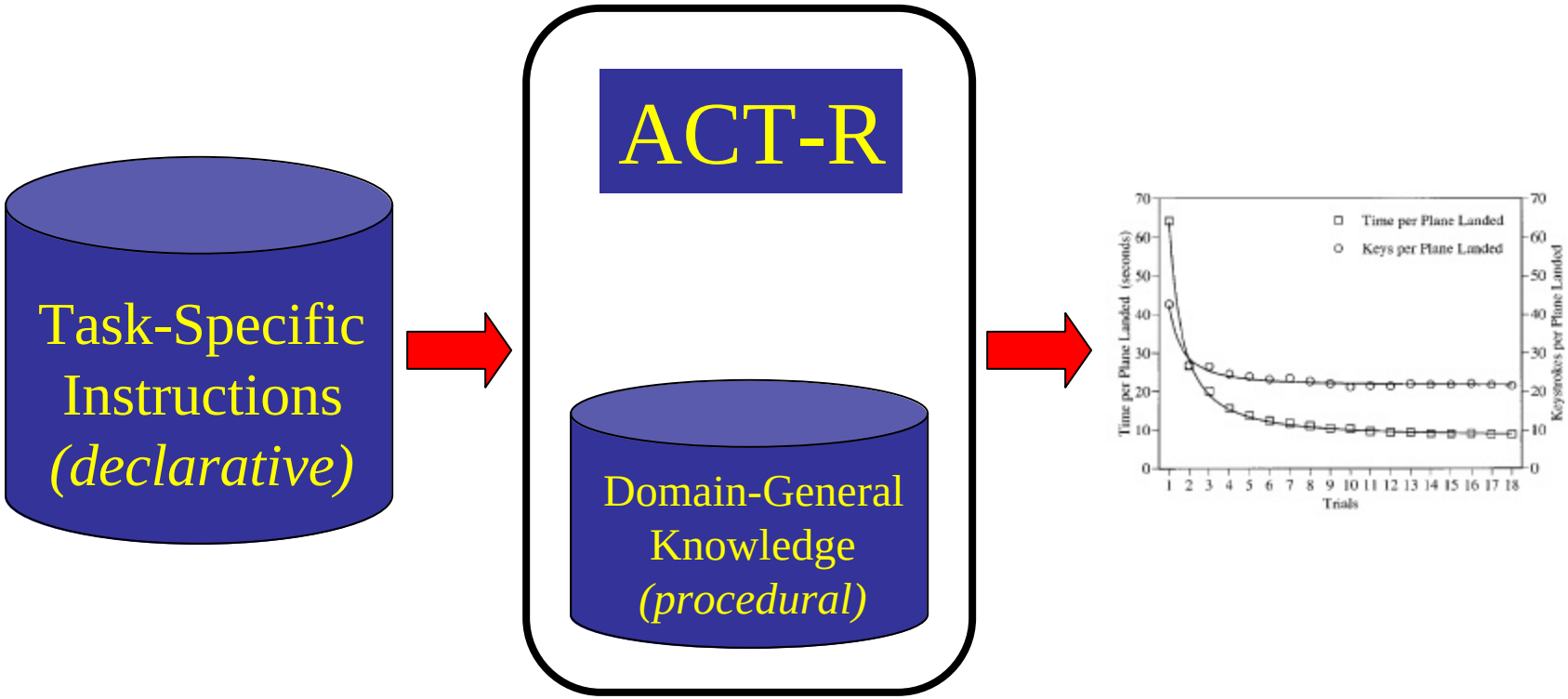
- Lee & Anderson (2000)
  - 2000 ICCM Conference: Model of expert performance using ACT-R/PM 1.0 and the experiment-ATC task
- Lee (2000)
  - Ph.D. Thesis: Hinted at (*and implemented on the side*) a model of learning where to look (i.e. locations) using ACT-R/PM 1.0 and ACT-R 5.0 and the experiment-ATC task
- Anderson (2000)
  - 2000 ACT-R Workshop: Model of learning from instructions using ACT-R 4.0 and a pseudo-ATC task
- Taatgen (2001)
  - 2001 ICCM Conference: Model of individual differences in learning using ACT-R 4.0 and a pseudo-ATC task

The Goal of the New Model

- Goal
  - A complete model of learning from instruction to expertise in the KA-ATC Task.

- Question
  - Will ACT-R 5.0 suffice?
    - Will production composition suffice?

The Model: Elegance is Virtue



Examples of Instructions

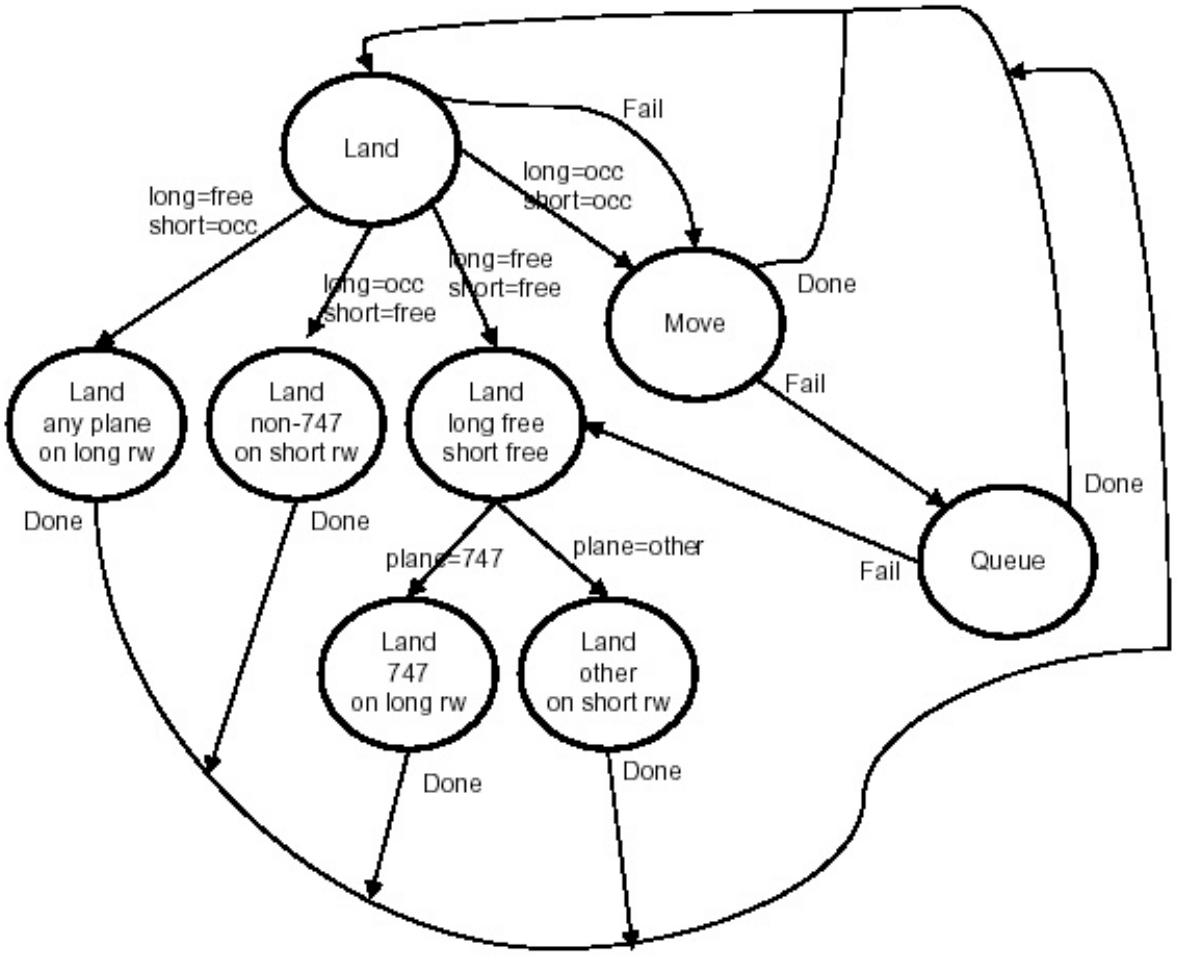
```
(land2
     isa instruction
     task land
     action scan
     arg1 wind-direction
     arg2 none
     prev land1)
```

```
(land3
     isa instruction
     task land
     action remember-string
     arg1 loc1
     arg2 none
     prev land2)
```

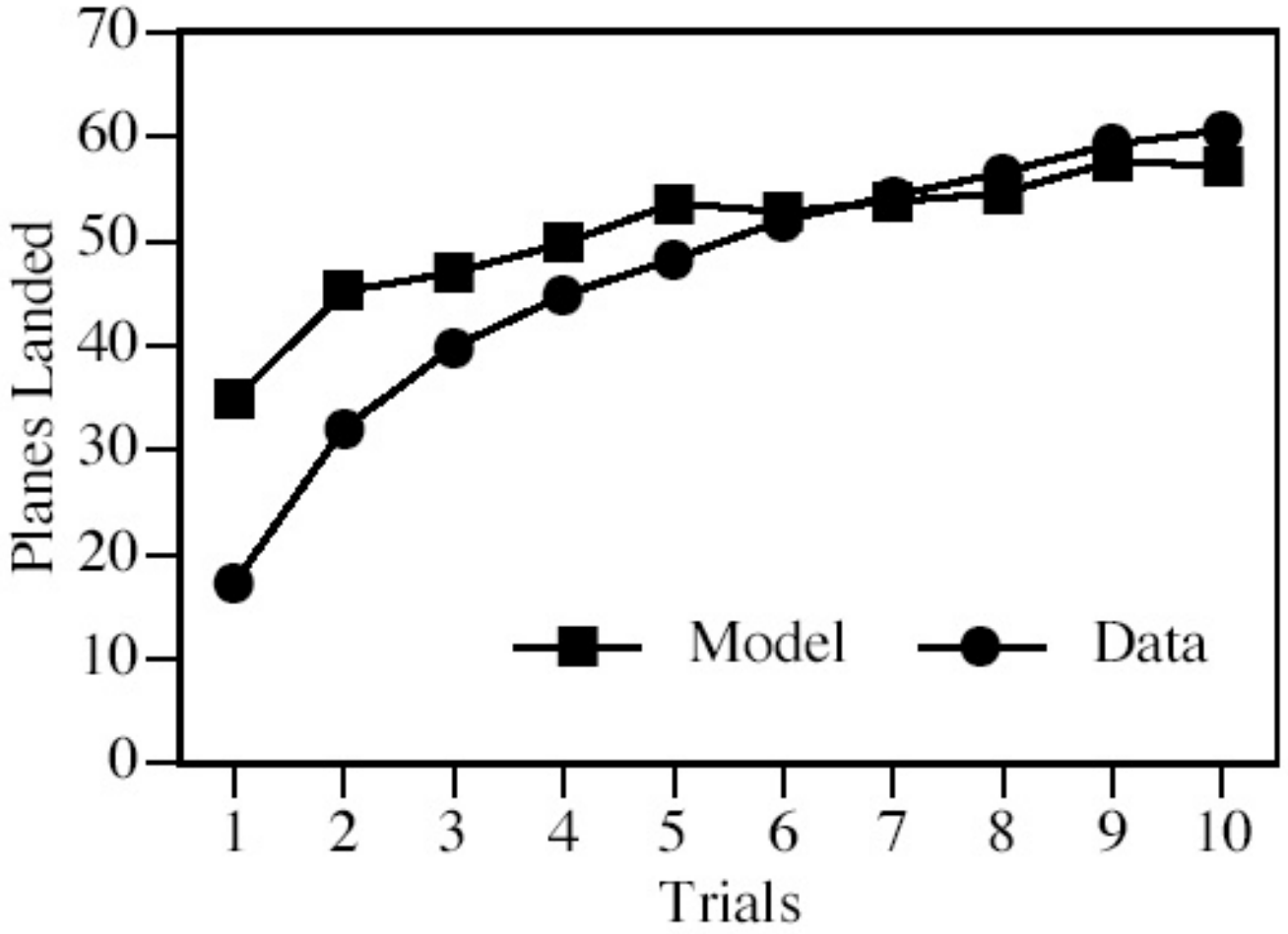
Task Independent Procedures

Operation	Description
Scan Scan-next Scan-empty	These scan operations move visual attention to a certain region on the screen, focus on an object in that region, and retrieve that object. Scan-next will only focus on objects that have not been recently attended, and Scan-empty will focus on empty space instead of text (necessary to find empty slots in the Hide-level)
Remember-loc Remember-string Remember-status	The remember operations store information from the current visual location or object in the global memory. Remember-loc stores the current visual location. Remember-string appends the currently attended word and stores it in the global memory. Remember-status parses a runaway-string and stores its status (free or occupied) in the goal
Press-enter Press-F1	Operations to press the enter and F1 key, respectively
Move-to-loc	Operation to move the cursor to a specified position on the screen by repeated pressing of the arrow keys
Compare-restart	Compare whether the two arguments of the action are equal. If this is the case, restart the present goal, else continue. This operation is used in combination with Scan-next to find a non-747 in Hide-level 1

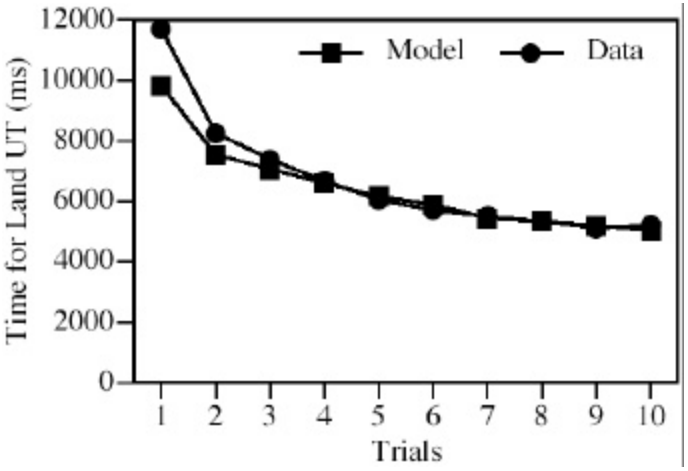
Model's Goal Control Structure

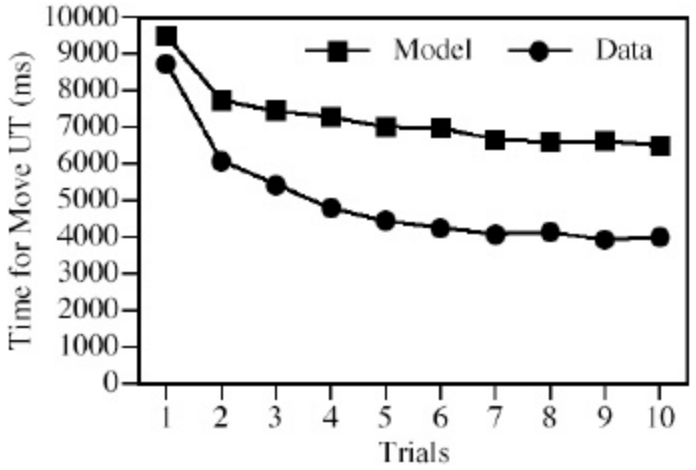


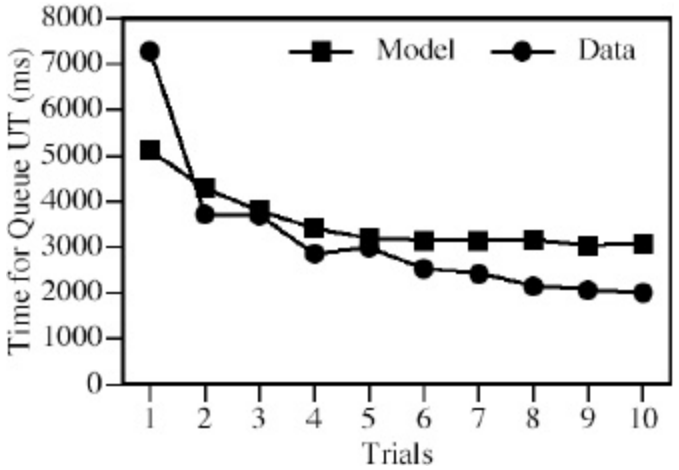
Task Level


70
60
50
40
30
20
10
0
Planes Landed
Model
Data
1
2
3
4
5
6
7
8
9
10
Trials

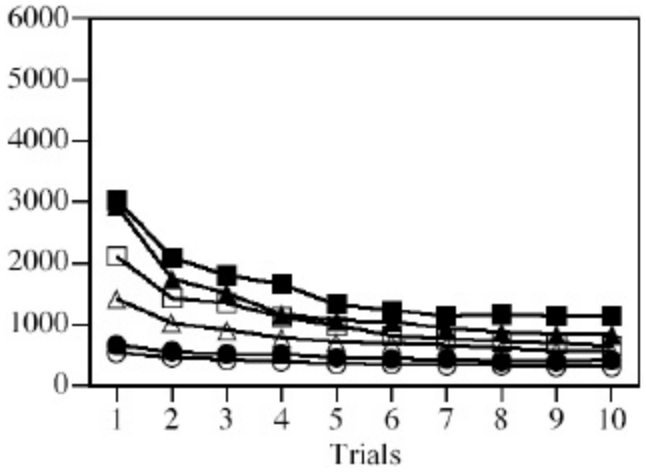
Unit Task Level



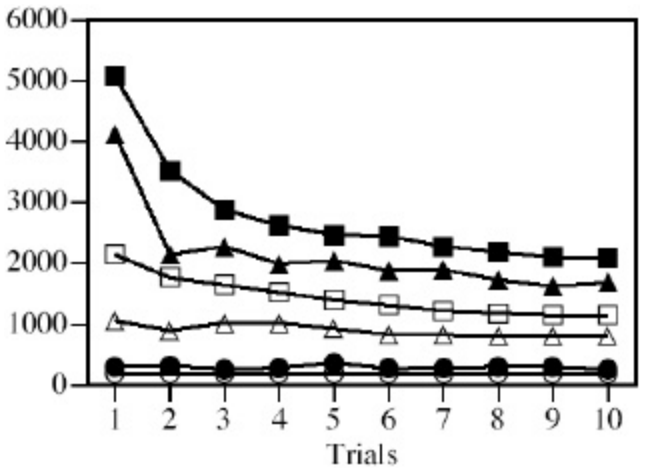




Functional Level



Data



Model

Yes, model is slower, but this is probably due to the conservative nature of production composition.

The Conclusion

Act III

Pluses

- It works!
- Uses default ACT-R 5.0 parameters (no parameter optimizations performed)
- Uses the experiment-ATC task
- It's simple
  - Modest set of task instructions + simple set of tas-independent procedures = complex learning behavior
- Displays interesting strategy change (e.g. John & Lallement's (1997) opportunistic strategy)
- And have I mentioned that it works?

Minuses

- Still starts out with too much knowledge
  - e.g. goal control structures
- Lacks strategy variations
  - This may be due to starting out with too much knowledge
  - (Solution: include knowledge discovery phase)
- Too slow at the keystroke level
  - (Solution: optimize production compiler for RPM)

The End?